REMARKS

Claims 1-38 were originally presented in the subject application. Claims 1, 9, 13, 21, 25-27 and 35 were amended in an Amendment and Response to Office Action dated November 14, 2005. No claims have herein been amended, added or canceled. Therefore, claims 1-38 remain in this case.

The addition of new matter has been scrupulously avoided.

Applicants respectfully request reconsideration and withdrawal of the sole ground of rejection.

35 U.S.C. §103 Rejection

The final Office Action rejected claims 1-38 under 35 U.S.C. §103, as allegedly obvious over Davidson et al. (U.S. Patent No. 6,042,614), hereinafter "Davidson," in view of Li et al. (U.S. Patent Application Publication No. 2003/0056200), hereinafter "Li." Applicants respectfully, but most strenuously, traverse this rejection as it applies to the amended claims.

Claim 1 recites a method of facilitating debugging of transactions. The method comprises executing a transaction on one processor of a plurality of processors, the transaction having debug information attached to the transaction. The method further comprises requesting, by the transaction, a service on another processor of the plurality of processors. The attached debug information is passed with the transaction from the one processor to the another processor, eliminating a need for attaching the debug information at the another processor. The path of the transaction is not predefined to a controller of the debugging, and at least a portion of the debug information is used to automatically establish a new debug session at the another processor without intervention.

Applicants submit that neither reference, nor their combination, teaches or suggests, for example, passing information used to automatically establish a new debug session at another processor without intervention, as claimed.

Against this aspect of claim 1, the Office Action cites to Li's teaching regarding the logging of data from monitoring. However, Applicants submit that a "debug session" implies more than simply logging data (collecting information and writing to a file). Rather, Applicants submit that a debug session implies an interactive aspect. For example, Applicants submit that FIG. 2 of the present application in conjunction with the description thereof in the specification implies an interactive aspect with a developer. In addition, attached is a hard copy printout of a definition for "debugging" from whatis.com, a well-known information technology web site. The whatis.com definition also clearly indicates an interactive aspect regarding identifying coding errors. In short, Applicants submit that the Office Action goes beyond the teaching of Li (logging data) by interpreting the same as a debug session, since there is no interactive aspect to the logging in Li. Instead, as shown in Li at, for example, FIG. 1 and the description thereof, the logged data is simply placed in a database for separate, later analysis. Applicants submit this does not comport with the generally understood meaning of a debug session.

Therefore, Applicant submits that claim 1 cannot be rendered obvious over Davidson in view of Li.

Independent claims 9, 13, 21, 25-27 and 35 each include a limitation similar to that argued above with respect to claim 1. Thus, the remarks made with respect to claim 1 are equally applicable thereto. Therefore, claims 9, 13, 21, 25-27 and 35 also cannot be rendered obvious over Davidson in view of Li.

Applicants submit that the dependent claims are allowable for the same reasons as the independent claims from which they directly or ultimately depend, as well as for their additional limitations.

For example, claim 3 recites providing, by the controller to the one processor, at least part of the debug information, wherein the debug information is provided to the another processor independent of the controller. Against claim 3, the Office Action cites to Davidson at FIG. 10 and column 14, lines 50-64.

However, the client host is the controller in Davidson, and FIG. 10 and the cited description thereof make clear that a determination as to whether a dbx engine is running on the remote host is communicated to the client host by the server. Thus, nothing happens regarding a dbx engine on the remote host without the knowledge of the client host. Moreover, the cited section makes clear that when there is no dbx engine running on the remote host, it is the client dbx engine that initiates a request for one to be created. Thus, Applicants submit that nothing is passed to the remote host dbx engine independent of the client host.

In addition, Applicants submit that Li fails to remedy the above-noted shortcoming of Davidson regarding claim 3. The Global Causal Identifier of Li is passed between the stub and the skeleton. See Li at numbered paragraph 0059. Indeed, the Global Causal Identifier is merely logged and analyzed in a separate, later process. See Li at FIG. 8 and numbered paragraph 0144. Thus, Li does not teach passing debug information as claimed in claim 3, and used to establish a new debug session, but merely logs data for post-runtime analysis.

Therefore, Applicants submit that claim 3 cannot be rendered obvious over Davidson in view of Li.

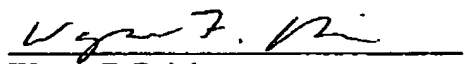
CONCLUSION

Applicants submit that the dependent claims not specifically addressed herein are allowable for the same reasons as the independent claims from which they directly or ultimately depend, as well as for their additional limitations.

For all the above reasons, Applicants maintain that the claims of the subject application define patentable subject matter and earnestly request allowance of claims 1-38.

If a telephone conference would be of assistance in advancing prosecution of the subject application, Applicants' undersigned attorney invites the Examiner to telephone him at the number provided.

Respectfully submitted,



Wayne F. Reinke
Attorney for Applicants
Registration No.: 36,650

Dated: March 07, 2006.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579